

Geometría Computacional

Ejercicio Práctico 3. Cierre Convexo: Graham Scan

Ailyn Rebollar Pérez
ailynarp12@ciencias.unam.mx

1 Objetivo

Se tiene como objetivo que los estudiantes recuerden e implementen el algoritmo de Graham Scan a partir de la clase Punto trabajada en el ejercicio práctico 1.

2 Implementación

Se proporcionará un esqueleto del archivo **cierreConvexo.py** y clase **CierreConvexo** que deberán implementar el cual está estructurado de la siguiente manera:

2.1 Atributos

- *puntos* : El conjunto de puntos para calcular su cierre convexo.
- *y* : La dirección que se tomará para encontrar el punto más chaparro.
- *chaparro* : El punto con menor coordenada y del conjunto.

2.2 Métodos

- **encuentraChaparro(self, d)**
Método que regresa el punto con la menor coordenada de acuerdo a la dirección dada.
- **ordenaPuntos(self)**
Método que regresa una lista de puntos ordenados de acuerdo a los ángulos polares formados entre el punto más chaparro y los dos siguientes puntos q y r.
- **grahamScan(self)**
Método que regresa una lista con los puntos que forman el cierre convexo usando el algoritmo de Graham Scan.

Nota para el método ordenaPuntos()

Lo complicado es relacionar los ángulos polares con los puntos correspondientes para ordenarlos.

2.3 Esqueleto del ejercicio

A continuación se presenta el esqueleto del ejercicio práctico de la clase CierreConvexo.

Listing 1: Archivo cierreConvexo.py

```
1 from punto import Punto
2 '''
```

```

3 Clase Cierre Convexo encargada de calcular
4 el cierre convexo de un conjunto de puntos
5 '''
6 class CierreConvexo:
7
8     '''
9     Constructor de la clase Cierre Convexo
10    @param puntos: el conjunto de puntos para calcular su cierre convexo.
11    @param d: la direccion que se tomara para encontrar el punto mas chaparro.
12    '''
13    def __init__(self, puntos,d):
14        self.puntos = puntos
15        self.d = d
16        self.chaparro = self.encuentraChaparro(d)
17
18    '''
19    Metodo que regresa el punto con la menor coordenada de acuerdo a la direccion ←
20    dada.
21    @param d: Indica que direccion tomaremos en caso de que haya dos puntos con la ←
22    misma coordenada y.
23    @return: el punto con menor coordenada y, si d = -1, entonces el mas chaparro ←
24    sera el punto con coordenada x menor, en caso de que sea 1, entonces se ←
25    toma el de mayor coordenada x.
26    '''
27    def encuentraChaparro(self, d):
28        return p1
29    '''
30    Metodo que regresa una lista de puntos ordenados de acuerdo
31    a los angulos polares formados entre el punto mas chaparro.
32    @return: la lista de puntos ordenados de acuerdo a los angulos
33    '''
34    def ordenaPuntos(self):
35        return []
36    '''
37    Metodo que regresa una lista con los puntos que forman el cierre convexo usando ←
38    el algoritmo de Graham Scan.
39    @return: la lista con los puntos que forman el cierre convexo.
40    '''
41    def grahamScan(self):
42        return []

```

3 Dudas

En caso de tener dudas específicas de su ejercicio que es **COMPLETAMENTE OPCIONAL**, podrán enviarlas por correo electrónico con el asunto **[Geometría Computacional]** incluyendo los corchetes a la dirección ailynrp12@ciencias.unam.mx donde se atenderán a la brevedad posible.